LEVEL III

12

AD A109453

Research Product 81-28

DESIGN GUIDELINES AND CRITERIA FOR USER/ OPERATOR TRANSACTIONS WITH BATTLEFIELD AUTOMATED SYSTEMS

VOLUME III-C:

HUMAN FACTORS ANALYSES OF USER/ OPERATOR TRANSACTIONS WITH ADMINISTRATION/LOGISTICS

HUMAN FACTORS TECHNICAL AREA

February 1981

U.S. ARMY RESEARCH INSTITUTE for the BEHAVIORAL and SOCIAL SCIENCES

U. S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES

A Field Operating Agency under the Jurisdiction of the Deputy Chief of Staff for Personnel

JOSEPH ZEIDNER
Technical Director

L. NEALE COSBY
Colonel, IN
Commander

Research accomplished under contract
to the Department of the Army

Synectics Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Research Product 81-28	2. GOVT ACCESSION NO.: AD-A109 453
3. RECIPIENT'S CATALOG NUMBER: Final rept.	
4. TITLE *(and Subtitle)*: DESIGN GUIDELINES AND CRITERIA FOR USER/OPERATOR TRANSACTIONS WITH BATTLEFIELD AUTOMATED SYSTEMS VOLUME III-C: HUMAN FACTORS ANALYSIS OF USER/OPERATOR TRANSACTIONS WITH ADMINISTRATION/LOGISTICS	5. TYPE OF REPORT & PERIOD COVERED: Interim: October 1979-February 1981
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Robert N. Parrish, Jesse L. Gates, Sara J. Munger	8. CONTRACT OR GRANT NUMBER(s): MDA903-80-C-0094
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Synectics Corporation, 10400 Eaton Place, Fairfax, VA 22030	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 2Q263744A793
11. CONTROLLING OFFICE NAME AND ADDRESS: US Army Research Institute for the Behavioral and Social Sciences, 5001 Eisenhower Avenue, Alexandria, VA 22333	12. REPORT DATE: February 1981
	13. NUMBER OF PAGES: 54
14. MONITORING AGENCY NAME & ADDRESS *(If different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Dr. Raymond C. Sidorsky, of the Human Factors Technical Area, ARI, is the Contracting Officer's Representative (COR) for this project.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Battlefield automated systems
Design criteria
Design Guidelines
Functional Standardization
Human-computer interaction
System analysis
Transaction Feature Analysis
User/operator transaction

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This document is one of a series in the Final Report of Phase I in a project to develop design guidelines and criteria for user/operator transactions with battlefield automated systems. The report is organized in five volumes as follows:

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

Item 20 (Cont'd)

I. Executive Summary (RR 1320)
II. Technical Report (TR 536)
III. In-Depth Analyses of Individual Systems
  A. Tactical Fire Direction System (TACFIRE) (RP 81-26)
  B. Tactical Computer Terminal (TCT) (RP 81-27)
  C. Admin/Log Automated Systems (this report)
  D. Intelligence Information Subsystem (IISS) (RP 81-29)
IV. Provisional Guidelines and Criteria (TR 537)
V. Background Literature (TR 538)

Volume I presents a succinct review of activities and products of the project's first phase. Volume II contains a technical discussion of the project's objectives, methodologies, results, conclusions, and implications for the design of user/operator transactions with battlefield automated systems. Volume III documents analyses of four unique battlefield automated systems selected to represent different stages of system development and different Army functional areas. Volume IV presents provisional guidelines and criteria for the design of transactions. Volume V provides a brief review of selected literature related to guidelines and criteria.

Research Product 81-28

DESIGN GUIDELINES AND CRITERIA FOR USER/ OPERATOR TRANSACTIONS WITH BATTLEFIELD AUTOMATED SYSTEMS

VOLUME III-C:

HUMAN FACTORS ANALYSES OF USER/ OPERATOR TRANSACTIONS WITH ADMINISTRATION/LOGISTICS

Robert N. Parrish, Jesse L. Gates, and Sarah J. Munger
SYNECTICS CORPORATION

Submitted by:
Stanley M. Halpin, Chief
HUMAN FACTORS TECHNICAL AREA

Approved by:
Edgar M. Johnson, Director
SYSTEMS RESEARCH LABORATORY

U.S. ARMY RESEARCH INSTITUTE FOR THE BEHAVIORAL AND SOCIAL SCIENCES
5001 Eisenhower Avenue, Alexandria, Virginia 22333

Office, Deputy Chief of Staff for Personnel
Department of the Army

February 1981

Army Project Number
2Q263744A793

Human Performance Effectiveness and Simulation

FOREWORD

The Human Factors Technical Area of the Army Research Institute (ARI) is concerned with helping users and operators cope with the ever increasing complexity of the battlefield automated systems by which they acquire, transmit, process, disseminate, and utilize information. Increased system complexity increases demands imposed on the human interacting with the machine. ARI's efforts in this area focus on human performance problems related to interactions with command and control centers, and on issues of system design and development. Research is addressed to such areas as user-oriented systems, software development, information management, staff operations and procedures, decision support, and systems integration and utilization.

An area of special concern in user-oriented systems is the improvement of the user-machine interface. Lacking consistent design principles, current practice results in a fragmented and unsystematic approach to system design, especially where the user/operator-system interaction is concerned. Despite numerous design efforts and the development of extensive system user information over several decades, this information remains widely scattered and relatively undocumented except as it exists within and reflects a particular system. The current effort is dedicated to the development of a comprehensive set of Human Factors guidelines and evaluation criteria for the design of user/operator transactions with battlefield automated systems. These guidelines and criteria are intended to assist proponents and managers of battlefield automated systems at each phase of system development to select the design features and operating procedures of the human-computer interface which best match the requirements and capabilities of anticipated users/operators.

Research in the area of user-oriented systems is conducted as an in-house effort augmented through contracts with uniquely qualified organizations. The present effort was conducted in collaboration with personnel from Synectics Corporation under contract MDA903-80-C-0094. The effort is responsive to requirements of Army Project 2Q263744A793, Human Performance Effectiveness and Simulation, and to special requirements of the U.S. Army Combined Arms Combat Developments Activity (CACDA), Fort Leavenworth, Kansas.

JOSEPH ZEIDNER
Technical Director

DESIGN GUIDELINES AND CRITERIA FOR USER/OPERATOR TRANSACTIONS WITH BATTLEFIELD AUTOMATED SYSTEMS VOLUME III-C: HUMAN FACTORS ANALYSIS OF USER/OPERATOR TRANSACTIONS WITH ADMINISTRATION/LOGISTICS AUTOMATED SYSTEMS

EXECUTIVE SUMMARY

Requirement:

To develop a comprehensive set of human factors guidelines and criteria for the design of user/operator transactions in battlefield automated systems for use by human factors specialists and system proponents, managers, and developers.

Procedure:

To provide data for a baseline functional description of user/operator transactions in battlefield automated systems, user/operator interactions in a series of systems were analyzed using a Transaction Feature Analysis technique. Data were collected during interviews with system experts and reviews of system documentation. Transactions were then compared across systems using a Transaction Compatability Analysis technique. Results of these analyses formed the data base for development of preliminary guidelines and criteria.

Findings:

An initial output of the preliminary review of systems was the following categorization of design features affecting user/operator transactions with battlefield automated systems: Control Methods, Display formats, Data Entry Assistance, Message Composition Aids, Data Retrieval Assistance, Glossaries, and Error Handling Techniques. Appropriate subcategories were established for each of the major design feature categories. Provisional guidelines were prepared for the following selected design feature topics: Command Methods for Alphanumeric Terminals, Selective Highlighting, and Information on Legal Entries. Guideline sets are organized around the following topics: Definition, Use, Applications, Types, Recommendations, and Advisory Comments. In addition, discussions are presented about each of the 34 subcategories of design features.

Utilization of Findings:

Findings from the analysis of individual systems may be useful to proponents in specifying user/operator requirements for future system evolution. In this project, the findings were incorporated in a data base on human factors requirements which provided the "real world" foundation for development of the provisional guidelines and criteria presented in volume IV of this report. The provisional guidelines and criteria will be utilized as the basis for development of the prototype handbook.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

SUMMARY

This document reports a human factors-oriented analysis of user/operator interactions with the DS4 Automated Run Book. Information about the system was gathered during a visit to the CSC Run Book development facility and from examination of program listings. Observations of the system were recorded with a Transaction Feature Analysis technique developed for this purpose. Transaction features analyzed with the technique were arranged by categories to facilitate presentation and discussion.

In general, the analysis indicated that the Run Book is a well-designed interface between the user and the DS4 software package. The analysis was limited in scope and depth by constraints in this project's charter and resources. Nonetheless, it suggests that from the user's point of view, there are no major problems or deficiencies in the system. No single feature was observed that by itself would be likely to degrade system performance significantly. However, a number of recommendations were offered that would help to make the Run Book an even "friendlier," easier-to-use system.

These recommendations are summarized in Table 1. The table is organized by the categories of design features described in the report. Each recommendation is evaluated, in the best judgment of the authors, in terms of hardware changes, software reprogramming, and changes in user performance. These evaluations cannot be expressed in quantitative terms because appropriate data could not be collected. Therefore, evaluations are expressed in terms of low (L), moderate (M), or high (H) impact on hardware, software, and performance with a minus sign indicating negative impact (cost) and a plus sign indicating positive impact (benefit).

Table 1

Summary of Design Feature Recommendations

and Their Impact on the System

CATEGORY	RECOMMENDATIONS	IMPACT* Hardware	Software	User Operator/ Performance
1. CONTROL METHODS				
1.1 Command Language	. Restrict use of command language to experienced users.	None	L-	L+
1.2 Menus	. Prevent menus scrolling off top of screen.	None	L-	L+
	. Move selection numbers closer to selection descriptions.	None	L-	L+
	. Arrange selection numbers with units digit aligned vertically.	None	L-	L+
	. Include command stacking capability.			
1.3 Function Keys	. Eliminate difference between "(a)" and " --" keys in data reduction function.	Not Known		M+
	. Provide ending transaction after last error corrected.			
1.4 Hybrid Methods	N/A			
1.5 Prompts/HELPS	***NO DEFICIENCIES	OBSERVED***		
2. DISPLAY FORMAT				
2.1 Fixed Alphanumeric	***NO DEFICIENCIES	OBSERVED***		
2.2 Variable-Length Alpha	N/A			
2.3 Graphic Displays	N/A			
2.4 Highlighting	. Use highlighting consistently.	None	L-	M+
3. DATA ENTRY AND HANDLING				
3.1 Information on Legal Entries	. Provide legal entry information in data reduction function.	None	M-	M+
3.2 Unburdening of Input	. Allow user to enter date as DDMMYY; have machine convert to Julian.	None	L-	L+
Verification of Input	. Eliminate menu-by-menu verification.	None	L-	M+
3.3 Interrupts and Work Recovery	. Provide restart capability.	None	M-	M+
	. Allow user to start error correction with menu on which error occurred.	None	L-	L+
3.4 Manipulating Stored Data	N/A			
4. MESSAGE COMPOSITION AIDS	N/A			

Table 1 (Continued)

CATEGORY	RECOMMENDATIONS	IMPACT* Hardware	Software	User Operator/ Performance
DATA RETRIEVAL ASSISTANCE	. Provide capability to retrieve a data record or block of records.	None	Unknown	L+ - M+
	. Provide capability to page backward through data records during data entry.	None	Unknown	L+ - M+
GLOSSARIES				
6.1 Standard Terms	. Delete personal promouns referring to system.	None	L- - M-	L+ - M+
	. Use terms consistently.	None	L-	L+
6.2 Character Sets and Labels	. Do not use "(a)" key to backspace.			
6.3 Glossary Availability and Use	DEVELOPMENT NOT SUFFICIENTLY ADVANCED FOR ANALYSIS			
6.4 Abbreviations and Coding	. Allow users to input abbreviations to make menu selections.	None	L-	L+ - M+
ERROR HANDLING				
7.1 Error Prevention	NO RECOMMENDATIONS			
7.2 Error Detection	NO RECOMMENDATIONS			
7.3 Error Feedback	. Provide explicit information on nature of error.	None	L-	L+ - M+
7.4 Error Correction/ Recovery	. Remove error message from screen after correction.	None	L-	M+
USER/OPERATOR CONFIGURATIONS	NO RECOMMENDATIONS			

INTRODUCTION

This document reports a human factors-oriented analysis of user/operator transactions with the Automated Run Book for the Direct Support Unit Standard Supply System (DS4). DS4 is "...a computer software package designed to operate in either a divisional or nondivisional environment as an aid to the manager in supply and stock control."[1] The Automated Run Book will provide a software interface between the user and the DS4.

As indicated above, the analysis focused on user/operator transactions. It therefore did not examine such traditional human engineering features as stroke width of characters, force-displacement characteristics of keys, color- or shape-coding of knobs and levers, control-display ratios, or arrangements of workplaces. Indeed, the analysis addressed both hardware and software only insofar as they affect user/operator transactions. Throughout the effort, the emphasis remained on transaction features such as command methods, display formats, data entry and handling, message composition, data retrieval, glossaries, error handling, and user/operator configurations.

This analysis of the Automated Run Book and those of other systems listed in the Preface, served to validate information gathered during an earlier survey of Army battlefield automated systems. It also provides additional information for a data base on user/operator transactions initially developed from the earlier survey. This data base identifies and classifies problems and deficiencies in the human-computer software interface of battlefield automated systems. It will provide the foundation for developing guidelines and criteria for the design of user/operator transactions with future systems.

No attempt is made here to integrate the analysis of the Automated Run Book with those of other systems. Such an integration clearly is required to permit the comparisons among systems that will reveal problems and deficiencies common to battlefield automated systems in general, and those unique to

---

[2] *Direct Support Unit Standard Supply System (DS4), Detailed Functional System Requirements (DFSR).* TM38-L32-2 (Test). Headquarters, Department of the Army, July 1976, p. 5-1.

a particular system. The integration of separate analyses, comparisons among systems, description of problems and deficiencies, and conclusions and implications drawn from results are reported in Volume II of the final report of this project's first phase.

Because the analyses are oriented toward validating and enlarging a data base of problems and deficiencies in battlefield automated systems in general, recommendations for changes to the Automated Run Book or any other particular system are not a major purpose of the effort. However, the analytical technique described later leads naturally to recommendations for resolving problems and deficiencies described by the technique, and these recommendations are discussed in connection with the analysis of the system's transaction features. This issue is discussed more fully later in the report.

OVERVIEW OF THE SYSTEM

BACKGROUND

At present, automatic data processing support to the supply function in direct support units is provided by DLOGS software running on the NCR 500 computer. This second-generation equipment currently is being replaced in non-divisional units by the newer Decentralized Automated Service Support System (DAS 3) computer. The supply function initially will be supported on the DAS 3 by PHOENIX, an interim software package consisting essentially of DLOGs programs modified to run on the new hardware. Later, PHOENIX will be replaced by the DS4, which will provide all the data processing services of DLOGS/PHOENIX, plus additional supply functions and inventory management features. The Automated Run Book is being developed at Fort Lee, Virginia by the Army's Computer Systems Command to provide a software interface between the DS4 user and the DS4 data processing programs.

PURPOSE AND MAJOR FUNCTIONS

Purpose

The purpose of the Automated Run Book is to assist functional personnel in using the DS4. It reduces the requirement to punch and handle cards during data entry and editing, and during preparations for running one or another of the DS4 processing cycles. Interacting with the Run Book online at a terminal, the user responds to menus or prompts to enter transactional data and/or parameters required for DS4 operations.

Major Functions

The Automated Run Book helps supply personnel to perform two major functions with the DS4: data reduction and production processing.

Data Reduction. Using the data reduction facility, the user can enter data into DAS 3 storage media. If the data relate to new transactions, they are entered directly into an input file from one of the user terminals. This file is then held for processing by one of the DS4 cycles. If the data originated as error cards from a cycle run earlier, they are first entered into a file from the magnetic tape unit or the card reader/punch. The Run Book's editing capability then helps the user to correct errors in the data, with the corrected file becoming input to a subsequent DS4 cycle.

Production Processing. This function helps the user execute DS4 cycles. It provides online capability to generate the Execution Control Language (ECL) that would otherwise require keypunching into cards. Interacting with a combination of menus and prompts, the user/operator specifies the parameters required to run a particular cycle. Using information entered by the user, the Run Book then constructs ECL control card images that invoke the desired cycle and pass to it the necessary control parameters. In one sense, this function is the DS4, from the user's point of view, since it relieves the user of the necessity to interact directly with the DS4 itself.

RELEVANT HARDWARE ELEMENTS

The Automated Run Book is a software package, and therefore does not itself incorporate any hardware. However, it will run on the DAS 3 computer,

which consists of a Honeywell Level 6 minicomputer containing the following major components:

a. CPU with 256K bytes storage.

b. Operator's console:

   1. Serial Printer

   2. Operator's terminal

c. 2 User terminals:

   1. Modified QWERTY keyboard

   2. CRT display--24 lines X 80 characters per line

d. Line printer.

e. Magnetic tape drive.

f. 3 disk drives--80 Mbytes each.

g. Card reader/punch.

Eventually, the DAS 3 will be distributed to all direct support and general support units now equipped with NCR 500 systems, to other active DS/GS units that are not presently automated, and to Reserve and National Guard units. Current expectations are that 193 DAS 3 systems will be fielded. Each system will be mounted in a 35-foot van, supplied with electrical power by 2 30KW generators.

RELEVANT SOFTWARE ELEMENTS

From the point of view of this project, the most important part of the Automated Run Book is the menu program, called Computer/Operator Dialogue Exchange (CODE). This program presents menus to the user, accepts and verifies his or her menu selections, provides additional information when the user is unsure about what to do next, and invokes other routines to perform the Run Book's major functions and to provide access to the user portion of the Honeywell Level 6 GCOS command language.

User interaction with the Automated Run Book begins when CODE presents a master menu (Figure 1). This menu confirms that the user is logged onto the system, and provides access to the Run Book's capabilities. Note in the

Figure that the user can ask for help by selecting option "0," or can terminate the session by selecting option "99." These two options will be

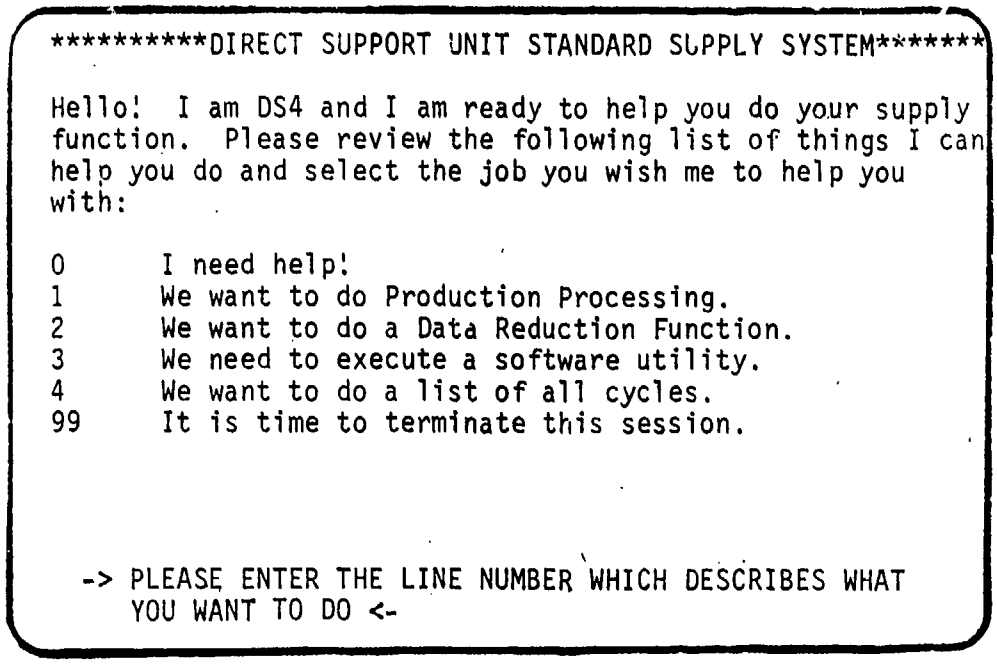

Figure 1. Master Menu for the DS4 Automated Run Book.

available to the user on every menu in the Run Book, and always with the same option numbers.

The master menu, as shown in the figure, permits the user to select either of the major functions described earlier, to invoke the Honeywell Level 6 GCOS command language (by selecting "software utility") for performing special functions, or to obtain a list of the 38 cycles provided in the DS4 (Table 2).

Production Processing

Selection of the production processing option starts the user on a path through a sequence of menus. These menus have been designed in a hierarchical structure, so that the system does not present extraneous information. The menus assist the user in specifying precisely the particular production process he or she desires to run. Figure 2 shows the major categories of these processes.

Table 2

List of the Supply Cycles Processed by the DS4

CODE	TITLE	FREQUENCY
DC	Daily Cycle	Daily
SC	Stock Status Report	Weekly
AO	DSU ASL lines with Dues Out	
ID	Due in File List Document Number Sequence	
IS	Due in File List Stock Number Sequence	
OD	Due Out File List Document Number Sequence	
OS	Due Out File List Stock Number Sequence	
CR	Customer Due Out Reconciliation	Semi-Monthly
AS	Authorized Stockase List	Monthly
BU	Bottoms Up Reconciliation	
DA	Demand Analysis (Demand History, OST/ASL Update)	
DS	Datalos Extract	
DH	Demand History Update	
FS	Financial Stockase List	
MR	Periodic MRO Statistics	
OU	Our Update Process	
SP	Supply Performance Report	
TR	Periodic Transaction Resister	
TS	Periodic Input Transaction Statistics	
UC	Catalos Update Process	
XP	Excess Process	
AP	Reportable Items Listing	
DB	Demand Analysis (PLL Comp - ASI/QSS Interconversion)	Quarterly
DP	Demand Analysis with PLL Computation	
SQ	Demand Analysis with ASL/GSS Inter-conversion	
PL	PLL List	
PU	PLL Update	
QC	QSS Listing	
GL	QSS Catalog	
XL	DX Listing	
SC	SSSC Catalog	
AR	ASL Replenishment (Stand Alone)	As Required
LS	Location Survey	
MS	Mass Cancellation Process	
PC	Parameter Change Process	
SI	Special Inventory	
UD	Unit Demand History Extraction/Insertion	
SX	SIMS-X	

```
========DS4 PRODUCTION REPORTS-PROCESSES========

0       I need help.
1       We want to do a DAILY report-process.
2       We want to do a WEEKLY report-process.
3       We want to do a SEMI-MONTHLY report-process.
4       We want to do a MONTHLY report-process.
5       We want to do a QUARTERLY report-process.
6       We want to do a AS REQUIRED report-process.
7       We want a LIST OF ALL REPORTS-PROCESSES.
99      It is time to TERMINATE THIS SESSION.



  -> Please enter the line number which describes what
     you want to do <-
```

Figure 2. The menu of major production processes available in the DS4.

The next menu to be presented depends on the user's selection from this menu. If the user, for example, selects the weekly report-process (option 2 in Figure 2), then CODE presents the menu illustrated in Figure 3. This menu

```
==============DSC WEEKLY REPORTS-PROCESSES============

0       I need HELP (FUNCTIONAL GUIDANCE)
1       We need to do a WEEKLY CONSOLIDATION
2 (SS)  We need to do a STOCK STATUS REPORT
3       We need to do a WEEKLY REPORTS PROCESS
99      It is time to TERMINATE this session.


  -> Please enter the line number which describes what
     you want to do <-
```

Figure 3. The menu of weekly reports-processes that are presented by the Automated Run Book.

data correction, or a combination of the two.

```
DAILY CYCLE DATA ENTRY/CORRECTION SELECTION

0   I need help!
1   We need to input new data.
2   We want to correct data.
3   We need a combination of 1 and 2 above.
99  It is time to terminate this session.


 -> Please enter the line number which describes what
    you want to do<-
```

Figure 5. The menu used to indicate whether the user wishes to enter new data or correct erroneous data for the daily cycle process.

Finally, the user must indicate the device from which data will be entered or corrected. For data entry, the device may be the user terminal, the magnetic tape unit, or the card reader/punch. For data correction, the original error cards may be entered from either magnetic tape or the card reader/punch; data corrections must be entered from the user terminal. Figure 6 illustrates the

```
PRODUCTION DATA ENTRY MEDIA SELECTION

0   I need help!
1   We need to enter data from this terminal.
2   We need to input data from card (CDR00).
3   We need to input data from tape (M 900)
4   We need a combination of 1 and 2 above.
5   We need a combination of 1 and 3 above.
6   We need a combination of 2 and 3 above.
7   We need a combination of 1, 2 and 3 above.
99  It is time to terminate this session


 -> Please enter the line number which describes what
    you want to do<-
```

Figure 6. The menu used to indicate which device(s) will be used for data entry during a data reduction process.

menu by which the user specifies the device(s) that will be used for data entry

Using the information obtained from the user's responses to the sequence of menus, the Run Book prepares the system to receive, in this example, input data. It also guides the user in the procedure required to complete the data entry operation. For example, if the user has selected card input, the system will provide instructions for readying the card reader/punch (Figure 7). When the user has followed these instructions and pressed the RETURN key, the Run

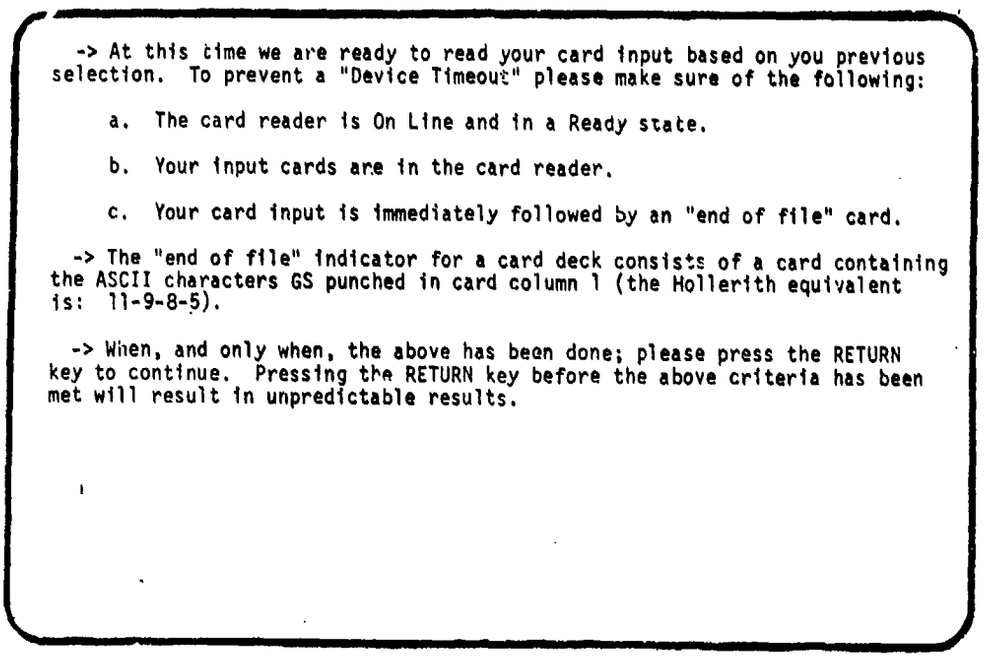

Figure 7. Online instructions for preparing the card reader for data entry.

Book invokes software to read the cards into an input data file. Finally, it returns to the master menu so that the user can select another function.

Software Utility

Selecting the software utility option from the master menu provides access to the user portion of the Honeywell Level 6 GCOS command language. CODE presents a single frame (Figure 8) providing a brief description of the command language format, and then waits for the user to enter a command (or to escape from the software utility by entering "QUIT").

```
**********HONEYWELL LEVEL 6 COMMANDS PROCESSOR**********

  -> At this time you may invoke any Honeywell Level 6 GCOS Command that is
not a "Operator Command".

  -> Commands are entered in the same format as specified in the Honeywell
Level 6 GCOS Commands Manual.

  -> Examples of some of the commands you may chose to do are:

     a.  CP full-pathname full-pathname

     b.  DP full-pathname[[ctl-ars1] [ctl-ars2] ....

     c.  CV full--athname[[ctl-ars1] [ctl-ars2] ....

     d.  TIME

  -> Once your command has been executed, you will be returned to the
previous screen.

ENTER COMMAND OR QUIT
```

Figure 8. CODE display frame explaining use of Honeywell Level 6 user command language.

The software utility probably will not be used heavily in the operational system; the developers of the Automated Run Book appear to have dealt with most of the operations that users will need to perform in using the DS4. Its availability nevertheless provides the flexibility required to cope with special situations or unexpected circumstances. This utility is discussed further later in this report, under "Analysis of Transaction Features."

ANALYSIS OF TRANSACTION FEATURES

The human factors analysis of the DS4 Automated Run Book is based on information gathered during a one-day visit to the development site at Fort Lee, and on inspection of a printout of the CODE program. During the site visit, development personnel provided a comprehensive demonstration of the Run Book in operation, and allowed the authors to experiment with the program at a terminal normally used for development activities. Observations taken during the visit to Fort Lee and the inspection of the CODE program listing were recorded using a Transaction Analysis technique developed for this purpose (Table 3 describes the technique). To facilitate the discussion of results that begins below, and also to facilitate comparisons among diverse systems, observations recorded with the technique were organized according to the categories shown in Table 4.

Table 3

Description of the Transaction Feature Analysis Technique

Transaction Feature. Describes the type of transaction being analyzed.

Description. Describes how the feature works in system operations. The description includes a specific example of the feature in straightforward, operational terms.

Behavioral Implication. Describes the feature's impact on the user's/operator's performance. The description includes what the individual must do--and must not do--in using the feature. It also includes requirements imposed upon the user/operator in terms of memory burden, error likelihood, skill requirements, and/or other performance-related issues.

Transactional Implication. Describes the feature's effect on the system's processing operations. The description includes issues such as the system's ability to detect errors, its error handling procedures, and/or the time required to complete transactions.

Consequences. Describes the feature's impact on overall system performance. Here, the analyst predicts the answers to questions such as the following: What effect does the feature have on the accuracy and timeliness of the data base? What effect does the feature have on the quantity and quality of output? Will the commander's picture of the battlefield be enhanced or distorted? Will targets be fired more quickly, or lost?

Recommended Resolution. Describes specific, detailed remedial action. These recommendations include changes to hardware, software, or procedures that will improve system performance

---

Table 4

Categories of Design Features Affecting User/Operator Transactions with Battlefield Automated Systems

1. CONTROL METHODS
   - 1.1 Command Languages
   - 1.2 Menus
   - 1.3 Function Keys
   - 1.4 Hybrid Methods
   - 1.5 Prompts/HELPS
2. DISPLAY FORMAT
   - 2.1 Fixed Alphanumeric Displays
   - 2.2 Variable-Length Alphanumeric Displays
   - 2.3 Graphic Displays
   - 2.4 Highlighting
3. DATA ENTRY AND HANDLING
   - 3.1 Information on Legal Entries
   - 3.2 Unburdening of Input
   - 3.3 Interrupts and Work Recovery
   - 3.4 Manipulating Stored Data
4. MESSAGE COMPOSITION AIDS
   - 4.1 System Design Features
   - 4.2 Format for Alphanumeric Messages
   - 4.3 Graphic Messages
5. DATA RETRIEVAL ASSISTANCE
   - 5.1 Query Method
   - 5.2 Query Structure
6. GLOSSARIES
   - 6.1 Standard Terms
   - 6.2 Character Sets and Labels
   - 6.3 Glossary Availability and Use
   - 6.4 Abbreviation and Coding
7. ERROR HANDLING
   - 7.1 Prevention
   - 7.2 Detection
   - 7.3 Feedback
   - 7.4 Correction/Recovery
8. USER/OPERATOR CONFIGURATION
   - 8.1 Operator(s) Only
   - 8.2 Operator(s) and User(s)
   - 8.3 Combined User/Operator
   - 8.4 User and Operator Chains

1. CONTROL METHODS

1.1 Command Language

The Automated Run Book uses the Honeywell Level 6 GCOS command language for special purpose operations. The language is invoked by selecting the SOFTWARE UTILITY option on the master menu.

The typical DS4 Automated Run Book user apparently will be a functional person trained in supply rather than computer operations or computer programming. GCOS is a powerful and highly useful computer language--for those who know the machine and the language well. It was designed for use by operators and programmers; preparing functional personnel to use it effectively will require considerable training and even more experience.

Unsophisticated users could easily become confused about proper usage of GCOS. Given the necessity to type in user commands, including delimiters and other punctuation, in a precise format, such a user might very well commit any of a number of errors, including:

a. A simple typographical error.

b. Leaving out a required parameter.

c. Entering an extraneous parameter.

d. Arranging parameters in the wrong order.

e. Entering incompatible parameters.

At best, such errors will result in error messages, delays in processing, and possibly user frustration. At worst, erroneous data could be entered into DS4 files, supply cycles could be processed unnecessarily or prematurely, or DS4 files could be destroyed.

To reduce the likelihood of these and possibly other undesirable consequences, access to GCOS should be restricted to users sufficiently trained to use it appropriately and effectively. This restriction could be accomplished by deleting the SOFTWARE UTILITY option from the master menu. Access to qualified users could be controlled by providing a means for the system operator to release GCOS to a user terminal, or by requiring the user to enter an alphabetic password instead of a numeric menu selection.

1.2 Menus

Menus provide the major method for selecting DS4 processing cycles, and for invoking the data entry and error correction functions. In general, the Run Book menus are very well designed from the user's point of view, with only minor deficiencies.

Menu Scrolling. One such deficiency is the method for presenting error messages. When a user selects an illegal option (for example, enters "5" from the master menu illustrated in Figure 1), the system responds with:

-> Only entries 0 through 4, 99 and HELP are valid selections <-

and then repeats its invitation to:

-> Please enter the line number which describes what you want to do <-

These messages are excellent in that they provide information about legal entries (see 3. DATA ENTRY AND HANDLING) and tell the user how to correct the error (however, see 7. ERROR HANDLING). The deficiency appears only if the user commits several errors on the same menu. Each time an error occurs, the error message and correction message are painted on the screen below the preceding messages. When the bottom line of the screen has been used, scrolling begins--and part or all of the menu might be lost off the top of the screen. This will happen, of course, at a time when the user still needs to be able to read the menu explanation and options.

One way to prevent this undesirable event would be to keep track of the progress of messages down the screen, and to repaint the menu when the bottom line was reached, placing the error and correction messages just below it. Another (and better) way would be to repaint the screen after each error, so that only the most recent set of messages would ever be displayed.

Menu Format. Another deficiency is the space between option numbers and the text description of the option in some menus (for example, the master menu; also, see 6. GLOSSARIES). This space is wide enough to require closer attention than should be necessary to associate an option number with its corresponding description. The width of this space could contribute to errors in entering menu selections (possibly exacerbating the problem described above).

Both double- and single-digit option numbers should be moved to the right, so that only one space separates numbers and descriptions.

Number Alighment. A related deficiency is the arrangement of single-digit option numbers vertically above the tens position of double-digit option numbers (for example, see Figure 9). This arrangement will not be a

```
==========DS4 MONTHLY REPORTS-PROCESSES==========

0       I need HELP (FUNCTIONAL GUIDANCE)
1       We need to do a MONTHLY CONSOLIDATION
2 (AP)  We need to do a REPORTABLE ITEMS LISTING (AESRS)
3 (AS)  We need to do a AUTHORIZED STOCKAGE LIST
4 (BU)  We need to do a BOTTOM UP RECONCILIATION
5 (CS)  We need to do a REQUEST FOR CATALOG DATA
6 (DA)  We need to do a DMD ANALYSIS (DHA EXTRACT, DMD HIST, OST, ASL UPDATE
7 (DH)  We need to do a DEMAND HISTORY UPDATE
8 (FS)  We need to do a FINANCIAL STOCKAGE LIST
9 (MK)  We need to do a PERIODIC MRO STATISTICS
10(OU)  We need to do a OUF UPDATE PROCESS
11(SP)  We need to do a SUPPLY PERFORMANCE REPORT
12(TR)  We need to do a PERIODIC TRANSACTION REGISTER
13(TS)  We need to do a PERIODIC INPUT TRNASACTION STATISTICS
14(CU)  We need to do a CATALOG UPDATE PROCESS
15(XP)  We need to do a EXCESS PROCESS
99      It is time to TERMINATE THIS SESSION


  -> Please enter the line number which describes what you want to do<-
```

Figure 9. Example of misaligned option numbers in a DS4 Automated Run Book menu.

serious source of errors. Even so, it could detract from operator "comfort" with the system, because it violates a population stereotype (i.e., most people in Western cultures have learned to expect that numbers will be listed with their units positions lined up vertically).

The recommendation offered above in regard to "menu format" would eliminate this deficiency also, of course.

Command Stacks. Finally, while menus normally are the preferred method for the types of operations for which they are used in the Automated Run Book, they have one disadvantage. That is, as users become more experienced, they often become bored and impatient with the necessity to step through a series of menus. Clearly, eliminating menus is not an acceptable solution to this problem. However, command stacks would be a solution that has been

used successfully in other systems. In this context, "command" does not refer to the GCOS command language. Instead, it refers to the capability to "stack" a sequence of menu selections on a single line.

For example, suppose a user frequently wishes to enter data from cards, as described in connection with the earlier discussion of data reduction, under "Relevant Software Elements." Suppose also that, from experience, this user has learned that the correct sequence of menu selections is "2" from the master menu (Figure 1), "2" from the data reduction cycle selection menu (Figure 4), "1" from the daily cycle data entry/correction selection menu (Figure 5), and "2" from the production data entry media selection menu (Figure 6). If the system had a command stack capability, the user would respond to the master menu by typing the following line:

2, 2, 1, 2

Encountering the first comma (or slash, or asterisk, or some delimiter), the system would "know" that the user had entered a command stack rather than a single menu selection. It would process each value just as though it had displayed and received selections from each menu, and then proceed to a verification recap. By "shortcutting" the menus (while still maintaining them for users who needed them), the system would reduce frustration for experienced users and save time in the bargain.

1.3 Function Keys

Although the user terminal is equipped with a variety of function keys, only the cursor control keys are used in the Automated Run Book. In this connection, two deficiencies were observed, both during the demonstration of the data reduction function. Both are potentially troublesome.

Cursor Movement (a). First, if an operator enters an erroneous character and then detects the error before leaving the data field, it is possible to correct the error. The first step is to move the cursor back to the error character, either by pressing the "@" key of the "<—" cursor control key (but not by pressing the "BACKSPACE" key, it acts like the "TAB" key). The next step is to press the key for the proper character, thereby overprinting the error character on the screen. However, what the user sees on the screen may or may not reflect what will go into the computer when the data entry is

completed and the "RETURN" key is pressed to enter the data. For example, suppose the user intends to type "YEH," inadvertently types "YEF," moves the cursor back to the "F," and types "H." On the screen, the user will now see "YEH," the proper character string. However, the character string that will be entered into the computer depends on how the cursor was moved backward. That is, if the user pressed the:

a. "@" key, the "H" will replace the "F" on the screen and in the input character string, so that the computer will receive "YEH."

b. "<—" key, the "H" will replace the "F" on the screen but not in the input character string, so that the computer will receive "YEFH."

Clearly, using the "<—" when attempting immediate correction of typographical errors will result in processing errors as well; time will be wasted, users will be frustrated, and errors may be introduced into the DS4 data base.

Unfortunately, this may well become a frequent problem in the field because the "<—" naturally lends itself to moving the cursor backward. This is especially true for personnel who have had experience on other systems. For this reason, the "<—" key should be modified to duplicate the operation of the "(@)" key.

Cursor Movement (b). In the error correction mode, correcting an error card begins with the system painting an 80-column image of the card near the top of the screen. The user can compare this image with the error card itself, on which have been indicated the data fields containing errors and the corrections to be made. If the Document Identifier Code (DIC) is wrong, it is corrected in the horizontally formatted card image. Then, to edit the remainder of the card, the user presses the "RETURN" key. The system breaks the horizontal card image into separate data items, with one item per line. Each line shows the card column(s) in which the data item appears, a field identifier that also serves as a prompt, and the data currently in that field.

The column numbers and field identifiers are protected; after the entire display is painted, the cursor returns automatically to the first character position of the data field on the second line (the first item--the DIC--was corrected, if necessary, on the horizontally-formatted card image). The user may either change the existing entry by typing in the correct data, or accept the existing entry by skipping the field. To advance to the next data field, the user may press any of four keys: "RETURN," "TAB," "BACKSPACE," or "↓."

The editing operation is not completed until the user either has entered correct data in the data field on the last line, or else has skipped past that field. Thus, if only the second field must be corrected in a transaction of, say, twelve fields, then the user must press "RETURN" (or "BACKSPACE," or "TAB," or "↓") ten times after correcting the error before he or she can proceed to the next transaction. While the necessity to do so probably will not increase user error rates, it does consume time and contribute to user boredom, frustration, and antipathy toward the system.

To correct this problem, the functions of at least three of the four keys named above should be modified. The "TAB" and "↓" keys could properly retain their present function of moving the cursor down to the next line when the user wishes to skip a field. The "BACKSPACE" key should perform the function it names; moving the cursor backward on a line. This would leave the "RETURN" key to signal the computer that the data editing procedure has been completed for a given transaction. These modifications would permit the user to proceed with editing operations much as they are performed now. However, when the last correction had been made, pressing the "RETURN" key would complete the transaction regardless of the current position of the cursor. The modifications would also have an ancillary benefit: they would provide different keys for different operations, thereby eliminating a source of confusion.

1.4 Hybrid Methods

Hybrid methods combine two or more command methods, such as using function keys to enter menu selections. No hybrid methods were observed in the Automated Run Book.

1.5 Prompts/HELPS

Prompts are used extensively in the Automated Run Book. Menu items, of course, provide explicit prompts for selecting functions. Questions provide prompts to elicit parameters required to generate ECL card images. Prompts are also provided in both data entry and error correction. In general, prompts appear to have been well-designed, providing clear and specific information about what is needed from the user.

HELPS provide additional information when the user is uncertain about how to proceed. At the time of the site visit, only a few HELPS had been implemented. These were quite good, being both explicit and clear. Other HELPS will be added as development continues; developers should be encouraged to show as much concern for the functional user in designing new HELPS as they have shown thus far.

2. DISPLAY FORMAT

2.1 Fixed Alphanumeric Displays

All displays observed during the site visit fit in this category. The only variable elements in the displays are the values entered into data fields; the fields themselves are of fixed length. Fixed alphanumeric displays are appropriate for the applications implemented in the Automated Run Book. They are generally well-designed to facilitate user interaction with the computer. No deficiencies were observed in this category (however, see 2.4 Highlighting).

2.2 Variable Length Alphanumeric Displays

The Automated Run Book does not employ this type of display, and apparently there are no plans to implement such displays in the future. Indeed, no evidence is known of any need for them.

2.3 Graphic Displays

The user terminal has a limited graphics capability. However, current applications of the Automated Run Book do not require graphics.

2.4 Highlighting

The user terminal has extensive highlighting capability: blinking, inverse video, two levels of brightness, boxing (using graphics features), and upper and lower case. The Run Book's developers have utilized some of these capabilities effectively, although not consistently. For example, consider Figure 10, which contains two examples of inconsistent highlighting.

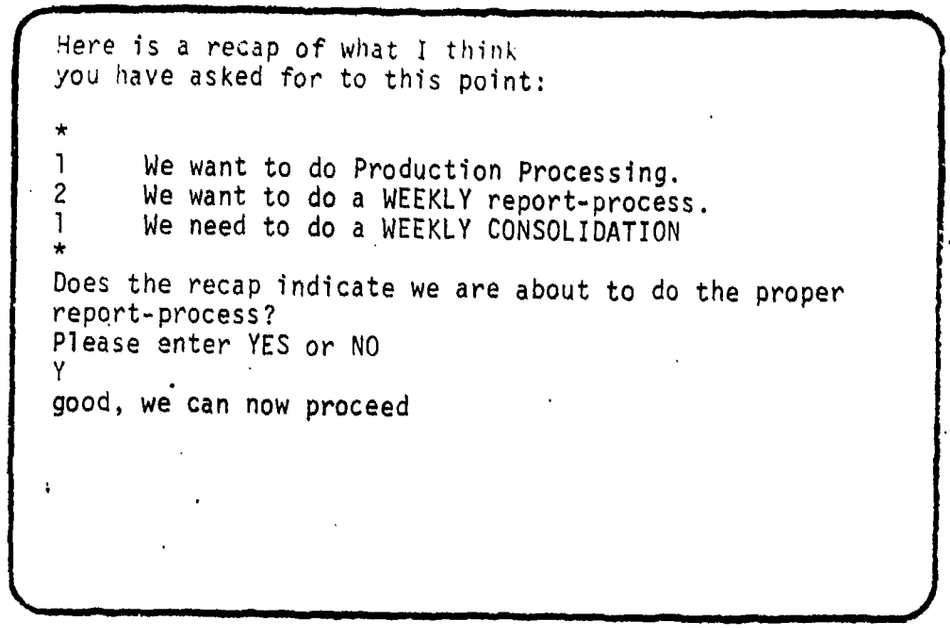

Figure 10. A sample of Automated Run Book output illustrating two examples of highlighting used inconsistently.

First, notice that sentences in the display begin with a capital letter, except for the last sentence, in which "good" begins with a lower case letter. Second, upper case letters are used to highlight important words in the display, such as "WEEKLY CONSOLIDATION," "YES," and "NO." But "Production Processing," surely of equal importance, is capitalized in the first letters only.

Similarly, in the data reduction function, prompts are displayed with lower brightness than data entries. However, the same highlighting procedure does not appear to be used in the production processing function when questions are used to prompt the user for ECL parameters.

Such inconsistencies (also see 6. GLOSSARIES) are not likely to be serious sources of error, nor are they likely to cause delays in data processing operations. However, even minor inconsistencies can introduce a jarring note into the user/computer relationship, adversely affecting the user's "image" of the system. That is, they can detract from the user's view of the computer as a well-designed, properly-functioning, reliable tool, thereby affecting the user's acceptance of the system.

User "image" and user acceptance are ill-defined and poorly understood issues in human-computer interaction, though some researchers believe they

may be critical to maximum system effectiveness. In any event, practicing consistency in the design of the user-computer interface surely does no harm, probably contributes to user comfort, and may even return unexpected benefits. In the case of highlighting, consistency can be achieved in two ways:

a. Always use the same type of highlighting for the same type of display situation. Repeating and extending practices already used in the Run Book:

   1. Use a capital letter to start each sentence, and appropriate punctuation to end it.

   2. Use upper case letters to highlight important words in prompts and messages.

   3. Use lower intensity in prompts and high intensity in user entries.

   4. Use blinking to highlight error feedback (see 7. ERROR HANDLING).

b. Always use that type of highlighting when that type of display situation arises.

3. DATA ENTRY AND HANDLING

3.1 Information on Legal Entries

In this category, too, there is some inconsistency in the design of the Automated Run Book. Menus, of course, provide information on legal entries automatically, because the set of options constitute the set of legal entries. In addition, when the user enters an erroneous menu selection, the sytem writes a reminder message below the menu containing the permissable values. However, in the data reduction function, no information on legal entries was observed during the site visit, either before the user's entry or following an error. In the latter case, the terminal merely emitted a "beep" to indicate an error, and the cursor returned to the first position of the data field.

Some form of legal entry information should be provided, at least following an error. It could be as simple as displaying a message saying "SEE TMXX-XXX-X FOR LEGAL DIC CODES." If nothing else, simply display a message such as "CHECK ORIGINAL DOCUMENT. IF YOU ENTERED (for example) THE DIC CORRECTLY, SET THE DOCUMENT ASIDE AND CANCEL THIS TRANSACTION. OTHERWISE, TRY AGAIN."

This at least does not leave the user to his or her own devices in figuring out what to do next.

3.2 Unburdening of Input

In general, the developers of the Automated Run Book have done a good job of relieving the user of requirements to perform tasks that the computer can do automatically, and of simplifying procedures that must be performed manually. Only minor deficiencies were observed in this category.

Entering Dates. During the demonstration of the Run Book, the user had occasion to enter a date, with a requirement to enter it as a Julian date. Many people are uncomfortable with Julian dates, even after considerable exposure. Therefore, unless the user can merely copy the date from a source document, the system should permit him or her to enter it in the conventional military format of DDMMYY. The machine could then convert this form to its Julian equivalent.

Verification of Input. After each menu selection in the production processing function, the Run Book asks the user to verify the selection. For example, if the user selects option 1 from the master menu illustrated earlier in Figure 1, the program responds:

```
You have selected:
1    We want to do Production Processing.
Please verify (YES or NO) your choice.
```

Assuming that this was indeed the intended choice, the user enters "YES" (or simply "Y"), and the program responds:

```
Thank you.
1    We want to do Production Processing.
```

It then displays the next menu in the sequence, the user enters a selection, and the verification process is repeated as described above. After all the menus in the series have been displayed and the user's selections have been entered and verified at each step, the system goes through yet another verification step. For example:

```
Here is a recap of what I think
you have asked for to this point.
*
1       We want to do Production Processing.
4       We want to do a MONTHLY report-process.
11 (SP)We need to do a SUPPLY PERFORMANCE REPORT
*
Does the recap indicate we are about to do the
proper report process?
Please enter YES or NO
```

If the user enters "NO" (or simply "N"), the program returns to the master menu. Otherwise, it replies:

```
good, we can now proceed
```

and then goes on to obtain parameters required to generate the report. This appears to be excessive verification for all but the most unsophisticated users. After attaining relatively moderate experience, users will probably find the menu-by-menu verification procedure irksome and unnecessarily time consuming, particularly when they know a recap will be presented after the last menu. Probably the best resolution for this deficiency would be to delete the menu-by-menu verification and retain the recap. Incidentally, the CODE program listing does not show a similar recap for the Data Reduction function: adding such a recap would be a desirable improvement to the Run Book.

3.3 Interrupts and Work Recovery

At present, capabilities are limited for interrupting a run currently in progress and restarting at a given point. For example, if a user started a particular DS4 cycle and then realized suddenly that a few transactions had been left out of the input stream, the only way to recover would be for the system operator to press the "CPU STOP" button on the CPU panel and then reinitialize the system. The user would then have to start the job over from the beginning.

If a user discovered an error in the recap described above under "Unburdening of Input," currently, the system would return to the master menu if the user entered "NO." This, of course, would require repeating the entire sequence of menus, rather than permitting the user to restart from the point of the error.

Development personnel have indicated that a restart capability is planned for implementation before the Run Book is fielded, so that the system need not be reinitialized when a job must be interrupted. They also plan to implement a provision to allow the user to return to the point of an error instead of returning to the master menu. In this regard, a desirable option for all menus, except the first one, would be: "98 Return to preceding display" or: "98 Back up. I want to change my last entry."

3.4 Manipulating Stored Data

The Automated Run Book does not manipulate stored data, except for menu entries and ECL parameters. Data manipulation is performed by DS4 processing cycles, controlled by parameters obtained from the Run Book. No deficiencies were observed in this category.

4. MESSAGE COMPOSITION AIDS

The Automated Run Book is not part of a message processing system. Therefore, this major cateogry and its subcategories are not applicable.

5. DATA RETRIEVAL ASSISTANCE

At present, the only data retrieval capability in the Run Book is the capability to obtain reports from various DS4 processing cycles. Developer personnel might wish to consider two possible enhancements:

a. Functional personnel might benefit from the capability to retrieve a given data record or block of records from a DS4 data file. Subject matter experts, of course, would have an opinion on this issue.

b. Users of the Run Book might benefit from the capability to page backward through transactions they have entered during data entry or error correction operations. This capability would allow them to review transactions when desired, before submitting them to a DS4 cycle.

6. GLOSSARIES

6.1 Standard Terms

The Automated Run Book appears to use standard terms throughout its menus and prompts. Therefore, functional personnel should not encounter unfamiliar terminology in relation to their specialities when they are introduced to the system. However, there are several issues related to terminology that could prove troublesome.

Personal Pronouns. One of these issues is ambiguous use of the first person pronoun. For example, in the master menu (recall Figure 1), the display begins, "Hello! I am DS4 and I am ready...". Here, the "I" refers to the system. But in the first option of the same menu, the "I" in "I need help!" refers to the user, rather than the system. While users in general will doubtless have little trouble making the distinction between references to the system and references to themselves, such ambiguity may introduce a discordant note into the user-computer interaction. Also, highly unsophisticated users, or those who feel intimidated by computers, might experience difficulty at least initially.

Probably more important, some users may react to the anthropomorphism of a machine that calls itself "I" or "me." Many if not most users would ignore this; others may feel amused, but some may regard it as dehumanizing, patronizing, or even insulting, particularly when the machine displays a message such as, "I will now allow you to choose another function." The evidence on this issue is primarily anecdotal, but it seems sometimes to be a factor in user acceptance of computer systems.

Related to this issue is the use of "we" in menus, as in "We need to do Production Processing." Some experts think that such terms promote a feeling in the user of partnership with the computer, of human and machine working together to accomplish a task. Others believe that a more appropriate approach is to promote a user's feeling of being in control of the interaction, with the computer serving merely as a tool for the human. Again, there is little evidence regarding this issue.

In the absence of clear evidence, perhaps the resolution to the issue of anthropomorphism is simply to avoid it. That is, the master menu could

begin, "Hello! This is the DS4 and it is ready..." Then, wherever personal pronouns refer to the computer or its software, they could be replaced with "DS4" or "it." Similarly, "we" would be replaced with "I." Thus, all personal pronouns would clearly refer to the user, avoiding any ambiguity or tendency to provoke a negative reaction.

Inconsistency. A minor inconsistency was observed in the wording of options in various menus. The following recap, discussed earlier in another context, provides an example:

```
Here is a recap of what I think
you have asked for to this point.
*
1       We want to do Production Processing.
4       We want to do a MONTHLY report-process.
11 (SP) We need to do a SUPPLY PERFORMANCE REPORT
*
Does the recap indicate we are about to do the
proper report process?
Please enter YES or NO
```

Notice that in the first two selections, the wording is "We want..." while in the third, the wording is "we need..." As noted earlier (see 2. DISPLAY FORMAT), such inconsistencies are not likely to influence error rates significantly or to cause excessive processing delays, at least for even moderately experienced users. Nontheless, some unsophisticated users could interpret these inconsistencies as deliberate features of the system design and become concerned about what they may be missing in, for example, the difference between "want" and "need." In any event, the greatest impact of such inconsistencies is likely to be in the area of the user's "image" of the system, as discussed earlier in this report.

6.2 Character Sets and Labels

Character sets and labels in the Automated Run Book are relatively standard. Except for using the "@" key for backspacing to correct a typographical error during data reduction, no deficiencies were observed in the use of particular characters or labels. One possible oddity was noticed during inspection of the CODE program listing. This was the difference between the zero and the capital "O." Most typewriters and printers have a zero that is narrower than the capital "O;" on the DAS 3 printer, the opposite is true.

6.3 Glossary Availability and Use

Indications from developer personnel indicate that HELPS to be provided by the Logistics Center (LOGCEN) will include glossary definitions for display online. If these definitions are not included in the LOGCEN's HELPS, developers should consider asking for them, since such materials greatly reduce the user's need to refer to offline documents.

6.4 Abbreviations and Coding

In general, the Automated Run Book uses abbreviations and codes in the data reduction function, but not in the production processing function. Even in data reduction, abbreviations and codes are used only in data fields of transaction card images (e.g., in the DIC field). In one respect, not using abbreviations and codes is unfortunate, since many of them presumably become well-learned by functional personnel. For example, Figure 11 shows the

```
========DS4 AS REQUIRED REPORTS-PROCESSES=========

0      I need HELP (FUNCTIONAL GUIDANCE)
1 (AR) We need to do a ASL REPLENISHMENT (STAND ALONE)
2 (LS) We need to do a LOCATION SURVEY PROCESS
3 (MC) We need to do a MASS CANCELLATION PROCESS
4 (PC) We need to do a PARAMETER CHANGE PROCESS
5 (SI) We need to do a SPECIAL INVENTORY
6 (SX) We need to do a SIMS-X PROCESS
7 (UD) We need to do a UNIT DEMAND HISTORY EXTRACTION AND INSERTION PROCESS
8      We need to do a CYCLIC ERROR LIST
9      We need to do a EDIT-ARRANGE ABEND SORT 1
10     We need to do a EDIT-ARRANGE ABEND SORT 2
99     It is time to TERMINATE THIS SESSION

  -> Please enter the line number which describes what you want to do <-
```

Figure 11. The menu for As-Required reports-processes in the DS4 Automated Run Book.

menu for As-Required report-processes. Notice that seven of the 12 options in the menu have a code associated with them. Developer personnel indicated during the site visit that these codes are standard in the supply function, and that functional personnel eventually learn them.

Functional personnel should have the option to use these codes instead of line numbers to indicate their menu selections. This capability would be particularly useful in connection with the ability to use command stacks (see 1.2. Menus). Given these capabilities, a user who has learned a particular sequence of menu selections and who knows the appropriate codes could enter these codes in a command stack instead of stepping through the menu sequence. Of course, to make these capabilities maximally effective, codes would need to be devised in all menus for options that do not currently have them. For example, "NH" might represent "NEED HELP," "ES" might represent "END this SESSION" (instead of "TERMINATE this session), "CE" might represent "CYCLIC ERROR LIST," and so on.

7. ERROR HANDLING

7.1 Error Prevention

The Automated Run Book incorporates some good error prevention techniques. In the production processing function, the recap at the end of a menu selection sequence should reduce the probability of invoking DS4 cycles inappropriately. The use of menus, of course, reduces errors; because they display all legal values, they reduce the memory burden on the user. The use of questions to elicit ECL parameter information helps to prevent entering the wrong kind of data. For example, asking for a date relieves the operator of the necessity to remember that, say, Data Field Number 1 requires a date rather than, say, a stock number. Also, in the data entry mode of the data reduction function, underlines are used to indicate the length of each data field, with each underline being replaced by the input character as data entry proceeds.

Indicating field length in this manner is useful in two ways:

a. It cues the user as to the type of data to be entered (the user soon learns, for example, that stock numbers are longer than DICs).

b. If the user inadvertently omits one or more characters, the presence of underlines at the end of the field provides a cue to review the data field and correct the error before entering it.

7.2 Error Detection

Again, the Automated Run Book incorporates good techniques. The use of range checks, legal value checks, and cross-field checks wherever possible greatly reduces the probability of errors contaminating the DS4 data base. In addition, the program checks each field as it is entered, rather than waiting for the entire transaction to be completed before beginning error checking procedures. This feature is particularly good, because it provides an immediate opportunity for the user to correct each error.

7.3 Error Feedback

This category is the weakest feature of the Automated Run Book's error handling features. In general, error feedback consists of an audible "beep" from the terminal, and then a recovery message. For example, if the user enters a "5" from the master menu, the system provides this response:

->Only entries 0 through 4, 99 and HELP are valid selections<-

It does not tell the user what the incorrect entry was, leaving him or her to determine what went wrong. While making this determination might not always be difficult, the system would be more helpful if it provided explicit feedback, e.g.,:

->You entered 5.
->Only entries 0 through 4, 99 and HELP are valid selections<-

The feedback would be even more helpful if it (e.g., the "5") were highlighted by blinking, leaving no doubt in the user's mind as to the nature of the error.

7.4 Error Correction/Recovery

As noted above and elsewhere in this analysis, error correction and recovery are handled quite well in the Automated Run Book. The only deficiency noted in this regard was observed in the data reduction function. When the user commits an error during data entry or error correction, a message is presented at the top of the display. This message is not removed from the screen after the user corrects the error; it remains in place until the current transaction is completed. This feature has at least two disadvan-

a. If the user sees the message, corrects the error, and then goes on to enter date in subsequent fields, he or she may look up, see that same message, and try to relate it to the current data field. Thus, the message could be a source of difficulty for the user, and an unnecessary source of frustration as well.

b. If the user commits a subsequent error, the second message merely overprints the first. If the second message is shorter than the first, the remaining "tail" of the first message could in effect change the meaning of the second message, or render the second message uninterpretable.

For both these reasons, these error messages should be cleared from the screen as soon as the user has entered the correction, perhaps by overprinting a line of blanks.

8. USER/OPERATOR CONFIGURATIONS

The DAS 3 computer will be operated by a system operator. The operator's interactions with functional users evidently will be minimal, particularly in regard to performing tasks related to the Automated Run Book and DS4.

Two users will be able to interact with the Automated Run Book at a time, one from each user terminal. However, each will be concerned with particular tasks, which will not necessarily be related to each other. Therefore, little interaction can be expected to occur between the users during DS4 operations.

CONCLUSION

In designing the Automated Run Book, system developers consciously set out to produce a smooth, "friendly," easy-to-use software interface between functional supply personnel and the DS4. Although limited in scope and depth, the analysis discussed in the preceding section shows that the developers succeeded admirably, in the main. Only minor deficiencies were observed in the Run Book's design features affecting user-computer interactions. Taken individually, none of these deficiencies could reasonably be expected to contribute significantly to errors in performance or to delays in data processing operations.

Even so, system developers should not ignore these deficiencies, for two reasons. First, alleviating or eliminating even minor deficiencies results in a higher level of user acceptance of the system, and a lower level of dissatisfaction and frustration for the user. Second, the cumulative effects of minor or even trivial deficiencies in the human computer interface could be significant to overall system effectiveness. The research literature does not explore such effects, so there are no data on how serious the accumulation of small deficiencies are. Nonetheless, though the risk of ignoring this issue cannot be stated precisely, that risk should be taken into account in future development of the Automated Run Book.

Finally, as noted earlier, the DAS3 computer, the DS4 software package, and the Automated Run Book ultimately will be delivered to almost 200 Direct Support and General Support Units in active, Reserve, and National Guard components. In terms of the number of user terminals to be fielded and the number of personnel that will be involved, this will be among the Army's larger battlefield automated systems. Because it will be so widely distributed, and because it will support the critical supply function, the system will be an important one to the Army. As such, the DAS3/DS4/Automated Run Book system deserves a more thorough analysis than could be accomplished in a single one-day visit and examination of one program listing. Though the observations discussed in this report are believed to be relevant and useful, an exhaustive analysis was beyond the charter and the resources of this project. System developers should consider seriously sponsoring such an analysis. Equally important, developers should also consider seeking regular human factors participation in the continuing development of the Automated Run Book, to ensure that it remains a "friendly," easy-to-use system.

RECOMMENDATIONS

The most important recommendation of this report is that the Run Book's developers continue to keep the eventual user in mind as conscientiously as they have done thus far. A collateral recommendation, as noted above, is that human factors assistance be sought in future development of the system.

As mentioned in the Introduction to this report, specific recommendations for changes to the Automated Run Book or any other system are not a major purpose of this project. Even so, the Transaction Feature Analysis technique leads to a recommendation to resolve each design feature problem analyzed with it. Thus, the reader will find a recommendation above, under "Analysis of Transaction Features," for each deficiency analyzed with it.

It must be emphasized that neither the analysis nor the recommendations presented in this report take into account any hardware, programming, or documentation constraints inherent in the current configuration of the system that might explain deficiencies or preclude implementing recommendations. Indeed, the authors consciously ignored such constraints. For example, they are aware that Automated Run Book developers are constrained by availability of storage and by characteristics of the Honeywell FVORMS software package used in the data reduction. Nonetheless, they avoided attempts to make trade-off judgments.

ARI and Synectics are aware that such judgments on the part of outside observers too often overlook implications apparent to those who know the system well. They are aware also that limitations in project resources precluded the kind of analysis suggested above, which might have provided sufficient understanding of the system to permit informed suggestions on trade-offs. For these reasons, recommendations in this report are offered on the working assumption that the developer could easily implement any and all of them. This working assumption is made, however, in full knowledge that the developer is in the best position to determine the feasibility of implementing each recommendation immediately in the present configuration, or the necessity to defer it to a later generation of the system.